boosting to the plurality reward values for the respective arm; and selecting one arm from the plurality of arms which has greatest calculated aggregated reward value. The contents corresponding to the selected arm is output, for example on a display.

BRIEF DESCRIPTION OF DRAWINGS

[0016] The above and other aspects may become more apparent by describing in detail illustrative, non-limiting embodiments thereof with reference to the accompanying drawings, in which:

[0017] FIG. **1** is a block diagram illustrating a system of providing recommended content according to an exemplary embodiment.

[0018] FIG. **2** is a flow diagram illustrating a method of selecting contents according to an exemplary embodiment.

[0019] FIG. **3** is a flow chart illustrating a method of recommending contents according to an exemplary embodiment.

[0020] FIG. **4** is a flow chart illustrating a method of optimizing weights for each arm according to an exemplary embodiment.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0021] Exemplary embodiments will now be described in detail with reference to the accompanying drawings. Exemplary embodiments may be embodied in many different forms and should not be construed as being limited to the illustrative exemplary embodiments set forth herein. Rather, the exemplary embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the illustrative concept to those skilled in the art. Also, well-known functions or constructions may be omitted to provide a clear and concise description of exemplary embodiments. The claims and their equivalents should be consulted to ascertain the true scope of an inventive concept.

[0022] In an exemplary embodiment, an ensemble contextual multi-arm bandit method using linear program boosting to find optimal combination of existing methods, in order to improve long term reward of decision making/strategy selection through finding optimal combination of exiting methods. In additional to the ensemble recommendation method, one or more exemplary embodiments may provide:

[0023] 1) an LPBoost algorithm of the ensemble recommendation method for solving large scale problem,

[0024] 2) an efficient LPBoost algorithm in the online learning scenario, and

[0025] 3) a parallel computational structure for efficient implementation of the ensemble recommendation engine.

[0026] For example, a personalized recommendation system is usually formulated as a multi-armed bandit problem with context information. The recommendation engine may proceed in the following discrete trails t=1, 2, 3, . . . In trail t,

[0027] 1) A new user, u_t, comes into the system. Denoted by A_t the set of arms (article, news, strategies) the user may choose. For each arm $a \in A_t$, denoted by x_t^a the feature vector containing information of both the user u_t and arm a. x_t^a is referred as the context that can be used to predict the reward by arm a to user u_t.

[0028] 2) Based on the arm selection algorithm trained till the previous trial, an expected reward r_t^a for each arm $a \in A_t$ is calculated. Based on $\{r_t^a: a \in A_t\}$, an arm a* is picked by the arm selection algorithm and is recommended to the user.

[0029] 3) An action is taken by the user corresponding to the recommended arm a* with an actual reward $y_t^{a^*}$ observed. The algorithm then improves its arm-selection strategy with the updated new observation $(x_t^{a^*}, a^*, y_t^{a^*})$. Note that the new observation only influence the reward prediction by the arm selection engine for arm a*.

[0030] Commonly, the reward may be defined as follows:

[0031] $y_t^{a^*}=1$, when a user accept recommendation at time t

[0032] 0, when a user deny recommendation at time t

[0033] However, it is not limited thereto. In an exemplary embodiment, a method described hereinafter may also be used for an online recommendation with continuous rewards.

[0034] Generally, a recommendation engine is to train the arm selection algorithm used in STEP Two and Three described above so that the expected T trail reward, $[\Sigma_{t=1}^{T} r_{t, a_t}]$, can be maximized, where a_t represents the arm selected in trail t by the algorithm. Equivalently, the algorithm aims to maximize the click through rate (CTR) based on the reward definition above, or any continuous rewards scenario. To achieve this aspect, an arm selection algorithm in an online changing environment targets for a balance between exploitation and exploration. For exploitation, an algorithm relies on its past experience to recommend the optimal arm that leads to maximum predicted reward. For exploration, instead of recommending the arm with maximum reward based on historical information, an algorithm creates randomness to obtain users' feedback to various recommendation solutions so as to improve the algorithm training. Over exploitation will cause the system not to be able to adapt to the continuously changing online environment so as to reduce long term expected reward. Over exploration will cause the system not sufficiently take advantage of the existing information for best arm recommendation so that limit the short term reward.

[0035] The existing contextual bandit algorithms may provide different contextual-reward models, different sampling methods, various arm selection strategies, and different exploration-exploitation trade off solutions in order to achieve the goal to maximize the expected reward.

[0036] For example, a list of different multi-arm bandit algorithms by categories are described below. However, the list below is provided by way of an example only and not by way of a limitation. One or more exemplary embodiment may incorporate any bandit algorithm as a basic learner and apply the LPBoost recommendation method, which is described in greater detail below. The selection of a basic learner is not limited by the following list and is provided by way of an example only.

[0037] 1. Category—Unguided Exploration

[0038] A) ϵ-greedy method: randomly select an arm with probability ϵ, and select the arm of the largest predicted reward with probability $1-\epsilon$.

[0039] B) Epoch-greedy method: run exploration/exploitation in epochs of length L, in which one step of exploration is first performed, followed by exploitation in the rest trials before starting the next epoch.